Team 3176 White Paper

Title: Arduino LED / RoboRio Integration	Author(s): Martin Wilson, Jackson Hogan, Evan Sorrell, Elaina Perry, Jenna Fivecoat, Sean Rhodes, Andrew Knotek
Subteam/Function: Programming	Subject: Diagnostics
Date/Season: 17/18	White Paper Number: 1

Abstract

The paper outlines effective ways of controlling the LED’s via the Arduino IDE program and Arduino microcontroller, as well as controlling the lights from the RoboRio

Definitions

Arduino- Microcontroller the assigns values to the LED’s and is capable of powering and controlling them.

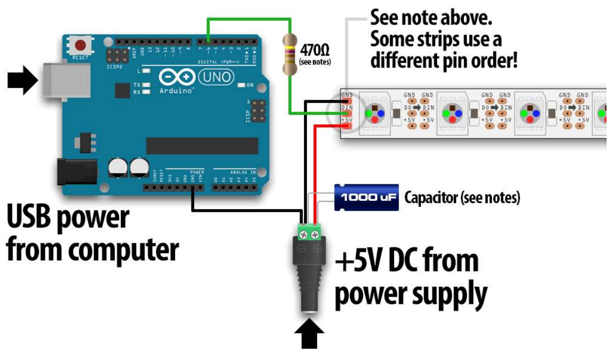
Arduino IDE- the program that will upload its code to the Arduino to control the LED’s

Wiring Outline

When connecting the wires be sure to

- Connect power from the power part of the NeoPixel strip to the positive end of the DC adapter (alternatively you can power the LED’s from the five volt pin on the Arduino)
- Connect a ground wire from the ground part of the NeoPixel strip to the DC adapter, and from the adapter to one of the ground pins on the Arduino microcontroller.

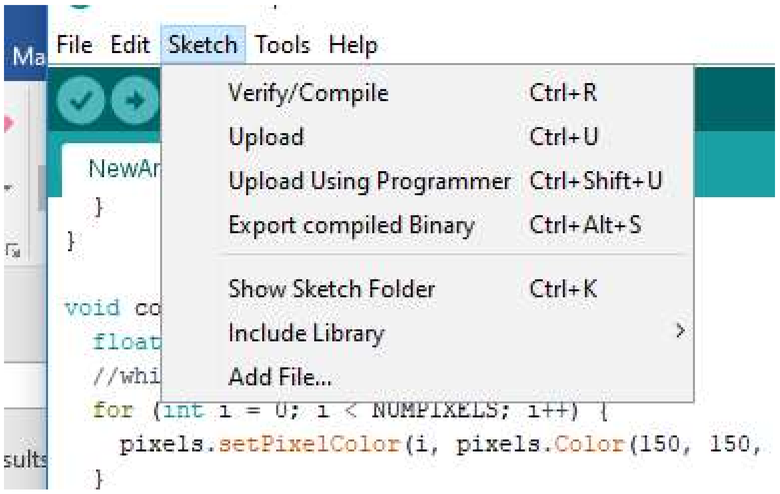
- Be sure to connect the 1000uF capacitor to the DC adapter if you use it (a diagram for a DC adapter show below)



- Connect the DI wire to a 470 ohm resistor and plug that into whatever PIN you assign in the Arduino IDE
- Connect ground first and disconnect ground last

Setting Up the NeoPixel Library

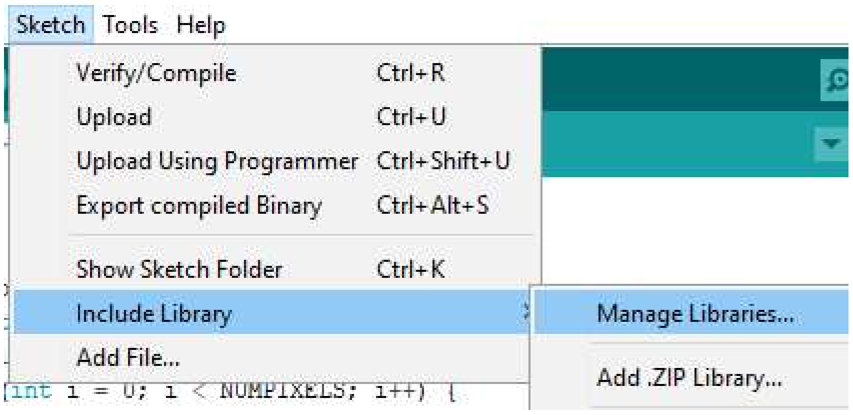
- The first thing to do when creating a NeoPixel program on a computer is to import the NeoPixel library with the following steps:
- Click the "Sketch" tab



File Edit Sketch Tools Help
Verify/Compile Ctrl+R
Upload Ctrl+U
Upload Using Programmer Ctrl+Shift+U
Export compiled Binary Ctrl+Alt+S
Show Sketch Folder Ctrl+K
Include Library
Add File...

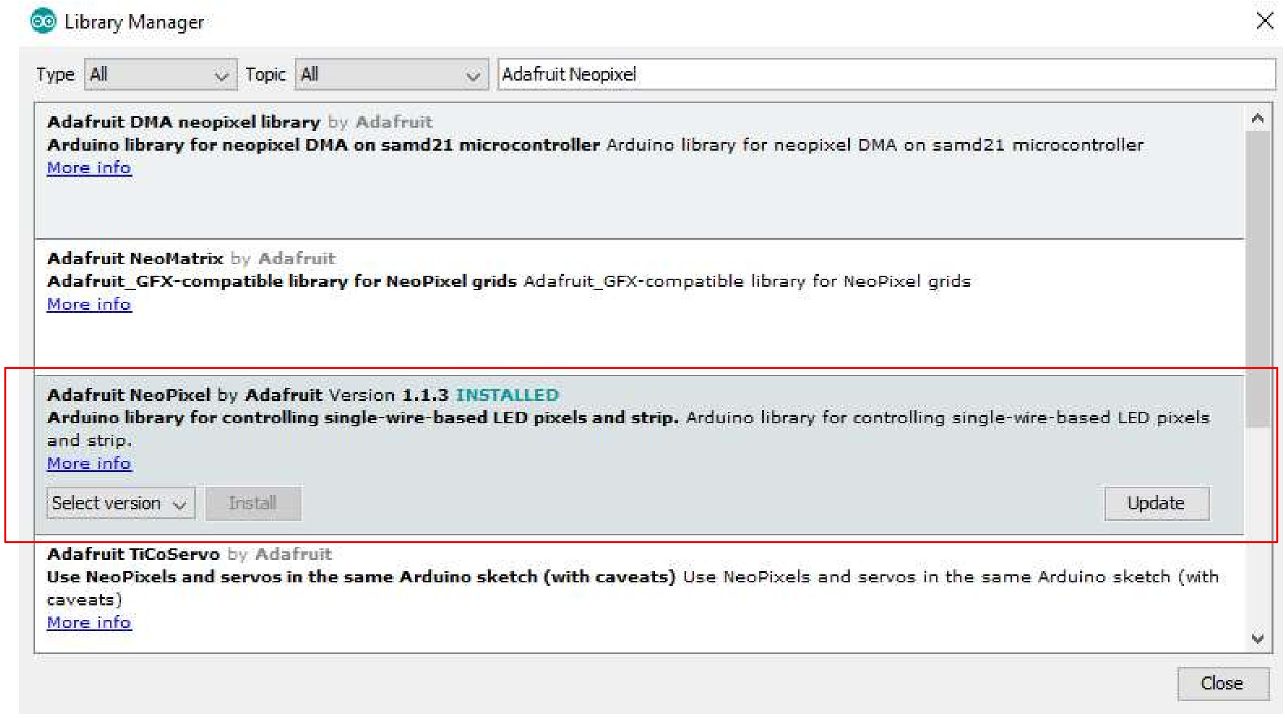
```
  }
}

void co
  float
  //whi
  for (int i = 0; i < NUMPIXELS; i++) {
    pixels.setPixelColor(i, pixels.Color(150, 150,
  }
```

- Mouse over “Include Library” and click “Manage Library”



```
int i = 0; i < NUMPIXELS; i++) {
```

- Search “Adafruit NeoPixel” and click on the library



Adafruit Neopixel

Adafruit DMA neopixel library by Adafruit
Arduino library for neopixel DMA on samd21 microcontroller Arduino library for neopixel DMA on samd21 microcontroller
More info

Adafruit NeoMatrix by Adafruit
Adafruit_GFX-compatible library for NeoPixel grids Adafruit_GFX-compatible library for NeoPixel grids
More info

Adafruit NeoPixel by **Adafruit** Version **1.1.3** INSTALLED
Arduino library for controlling single-wire-based LED pixels and strip. Arduino library for controlling single-wire-based LED pixels and strip.
More info

Adafruit TiCoServo by Adafruit
Use NeoPixels and servos in the same Arduino sketch (with caveats) Use NeoPixels and servos in the same Arduino sketch (with caveats)
More info

Important Methods and Practices for the Arduino IDE

- #define PIN X Defines a PIN to use on the microcontroller(Replace X with the pin you choose)
- #define NUMPIXELS X Defines a number of pixels X that are on the strip
- #ifdef __AVR__

```
#include <avr/power.h>

#endif
```

These lines initialize the language

This is an example of the initialization of those lines

```
#include <Adafruit_NeoPixel.h>

#include <Wire.h>

#ifdef __AVR__


#include <avr/power.h>

#endif

#define PIN           8

#define NUMPIXELS      150


Adafruit_NeoPixel pixels = Adafruit_NeoPixel(NUMPIXELS, PIN, NEO_GRB +
NEO_KHZ800);
```

- The following method is a replacement for the delay() method:

```
void wait(long x) {
  long currentTime = millis();
  long goTime = millis() + x;
  while (currentTime < goTime) {
    currentTime = millis();
  }
```

- The issue with the delay() method is that it pauses the processor so it cannot accept inputs. This method serves as a solution for this problem.
- This line gives each pixel an ID:

  ```
  Adafruit_NeoPixel pixels = Adafruit_NeoPixel(NUMPIXELS, PIN, NEO_GRB + NEO_KHZ800);
  ```
- This method sets a pixel to a color:

  ```
  pixels.setPixelColor(<pixelID>, pixels.Color(<Red Value>, <Green Value>, <Blue Value>);
  ```
- This method initializes the pixel library:

  ```
  pixels.begin();
  ```
- When attempting to light a strip, it is bad practice to light up LED’s line by line. Instead, use the following code segments to achieve the same effect in a cleaner way.

  ```
  //Every other LED beginning with the first
  for (int i = 0; i < NUMPIXELS; i += 2) {
     pixels.setPixelColor(i, pixels.Color(0, 0, 0));
    }


  //Every other LED beginning with the second
     for (int i = 1; i < NUMPIXELS; i += 2) {
     pixels.setPixelColor(i, pixels.Color(150, 0, 150));
    }
   pixels.show();
  ```

```
//Every LED
  for (int i = 0; i < NUMPIXELS; i++) {
    pixels.setPixelColor(i, pixels.Color(150, 150, 150));
  }
  pixels.show();


//Cycle through LED’s
  for (int i = 0; i < NUMPIXELS; i++) {
    pixels.setPixelColor(i, pixels.Color(150, 0, 150));
    pixels.show();
    wait(10);
    pixels.setPixelColor(i, pixels.Color(0, 0, 0));
    pixels.show();
  }
```